Chapter 2: Application Analysis

To ensure correct operation of HUD's application code after December 31, 1999, all programs need to be checked and possibly changed. Even programs which are expected to be dropped need to be examined. Contingency plans have to be developed in the event programs cannot be dropped as planned.

Once the inventory or working set of programs is established, it is analyzed to determine the work to be done and the sequencing of that work. Tools such as SystemVision 2000 may support this effort.

In many cases, the actual changes are relatively simple. The coordination and verification of the changes is more complicated. The policies for these changes must be documented and followed.

2. 1. Application Inventory of Programs

The first step to update an application is to produce an inventory of all its components. The initial list has to be checked to discard programs no longer in use. The source code set is compared to the load library (or Unisys absolutes, relocatables and object modules) to verify the correspondence between source code and the operating programs.

2. 2. Configuration Management

The development team must plan for a large volume of Year 2000 changes and the concurrent requirement to modify the application to meet changing customer requirement. This makes the tracking and coordination of changes a critical success factor in the conversion. Each renovation team must track all changes made, not only for troubleshooting, but also to provide information for possible audits. A rigorous configuration management process is therefore required to support:

- Change Control,
- Status Accounting,
- Component Identification,
- Physical Verification, and
- Functional Verification.

Configuration Management is covered in more detail in **Chapter 3**.

2. 3. Documentation

Media and consulting groups are already discussing the probable shortage of programmers over the course of the Year 2000 conversion. There is a considerable risk of high turnover during these projects. Documenting all information and procedures developed during the conversion process will significantly reduce the learning curve for new team members.

Many changes can be required to renovate all the dates in an application. Change control documentation prescribes these changes and when they are to be made.

Section 2.4.2 discusses alternative strategies for date fields. A decision is required for every field including a decision to do nothing. Each decision must be documented to remove all doubt about whether the field is being left unchanged or was just missed. The documentation of the planned changes is essential to keep everyone working together. All renovators need to be able to quickly find the strategy for each field. Good documentation—available to all—greatly improves the ability to complete the task.

Modifications to files and databases must also be recorded. There is a need to know what fields are being changed and how they are changing. This is especially important when more than one person, division, agency or other business partner is involved. Clear documentation on the layout of files and databases is essential for everyone working with those entities.

Once the planned changes are documented, the next step is tracking them as they are being accomplished. To fulfill management requirements , team members must be able to quickly determine who is working on a change and what the status of that change is. On-line documentation can disseminate to all team members the current status of various changes.

2. 4. Planning

HUD's plan for the Year 2000 conversion is based on a measure of lines of code changed as compared to the total to be changed. **Table 2-1** presents a timetable that was established based on corrections to lines of code.

Table 2-1: Timetable Based on Corrections to Lines of Code

Percentage To Be Complete	Date
20	9/30/1997
90	9/30/1998
100	12/31/1998

The Year 2000 changes require extensive analysis and coordination. Date field changes in files and databases frequently affect more than one program, division or business partner.

The work to be done must be analyzed, planned and documented so that all renovators know the policy for a given date field and the schedules to be followed. Any change in plans must be documented and all affected team members alerted.

2. 4. 1. Identify Date Field Changes

The first task is to produce a list of all the date fields in the system. HUD has acquired SystemVision 2000 for Hitachi and is examining tools for Unisys. ABSS has developed a tool to scan PC and LAN code. These tools perform an initial scan of the code and highlight most of the date fields.

2. 4. 1. 1. Transferred Files

Files are routinely transferred between programs within an application, between internal HUD applications and between HUD applications and those of its business partners (other government agencies and service providers). Any file containing dates must be analyzed and have a strategy developed (and agreed to) for each field and file record size. To control the renovation, changes must be documented where they can be retrieved by all the affected renovators. **Table 2-2** is a sample of the documentation required.

Table 2-2 : Date Field Control

File	Record	Item	Change	New Picture	Window*
FileA	RecA	Date-in	Y	CCYYMMDD	--
FileA	RecA2	Date-2	N	YYMMDD	45

* See the discussion of Windowing in Section 2.4.2.1

Changing date fields in files requires task coordination and scheduling between all who use the files, including HUD business partners. The identification of date fields must be done early in the process to allow for this coordination. If changes cannot be done simultaneously, bridges must be written. Bridges are routines which expand or contract batch data files to reconcile data format differences between the sending and receiving systems. Bridges can allow time to reprogram and change some of the date fields.

2. 4. 1. 2. Inputs/Outputs for Each Component

Entered data and displayed data may need changes to date fields. Reports and transaction screens must be examined to make sure the information is clear to both the human user and the computer code.

2. 4. 1. 3. Identify Shared Databases

Databases are almost always shared between programs and sometimes they are shared between applications. All the issues of transferred files apply here. The coordination issue is more complicated because the bridging technique cannot be used. The database is in either the old format or the new one. This makes the coordination of the changes more difficult.

There are other issues in changing databases. The entire database may have to be reloaded and reorganized. Issues include efficiency design considerations as well as scheduling.

2. 4. 1. 4. Special Flag Values

Look for hard-coded or literal dates used as default date values, or flags (for example, minimum, maximum, void). Null values, low values or high values must be used as the default value instead of any specified date. A particularly dangerous case is if 99, 99365, 99/12/31 or 00 is used to indicate an invalid date or end-of-input.

Note: Please report widespread use of any such default or special dates to the Year 2000 Technical Panel via the Team 2000 cc:Mail box. The Team 2000 project office serves as a central source of Year 2000 specific technical information for the HUD IT community.

2. 4. 2. Alternative Approaches

There are three major approaches for the conversion of date fields. See the **Standards for Year 2000 Conversion / Year 2000 Technical Panel, Document E** in the **Reference Library** for more details. The options are:

- **Windowing**. This technique requires coding and the assumption of a windowing time frame.
- **Encoding**. This uses a compression technique. Its primary value is in avoiding the need to expand databases and data files.
- **Field Expansion**. This is the actual expansion of the field to include the century.

2. 4. 2. 1. Windowing, Century Derivation and Coding

Windowing is a technique which retains the 2-digit date field, but modifies the application's programs to use a century-assigning technique to interpret each two-digit year field. A cut-off year is selected and any 2-digit year less than the value of that cut-off year is assumed to be in the 21st century.

For example, assume a certain variable can never have a date earlier than 1956; the code is written to assume that dates from 00 through 55 are actually 2000 through 2055. Values from 56 through 99 would be treated as 1956 through 1999. Different fields in an application may need different cut-off years.

Windowing techniques will not work for date ranges spanning over 100 years, though whether this is a problem or not depends on the range of the date calculation and on which windowing technique you use.

2. 4. 2. 1. 1. Two Windowing Techniques

There are two windowing techniques: *fixed* and *sliding*.

- **Fixed Windows.** The fixed window technique is based upon a fixed year. The year is set for the field and will never be changed. This assumes that the system will be terminated before the window becomes a problem.
- **Sliding Windows.** For the sliding window technique, the century of a date depends on current year. For instance, if the range is 30 years in the past and 69 years in the future, based on the current year of 1997, then the date window is 1967 through 2066. Next year, 1998, the window advances to include 1968 through 2067. Where possible, this technique provides a permanent solution to the Year 2000 Problem.

2. 4. 2. 1. 2. Problems With Windowing

The windowing approach saves time and costs initially, but leaves a complicated coding solution for future maintenance and support. The implementation of windowed coding is also more error prone because of the following:

- **Maintenance**. The application logic must include the windowing logic for each impacted date field. Ongoing maintenance becomes more challenging and more expensive.
- **File Transfer**. Consider two programs sharing a file. If one converts the format to a four-digit year and the other does not, a bridge is needed. If neither converts, they both have to use the same cut-off year for windowing.
- **Sorting**. Proper sequencing of years without a century is not possible once the data crosses the Year 2000 boundary. When dates have to be sorted or compared across centuries, the fields must be expanded. This could be done in the file or database. The alternative is to accommodate the century either logically or physically within the program.

2. 4. 2. 2. Compression and Encoding

This approach employs any of a number of schemes to compress or encode a century into a date variable without expanding its size. Records, files and non-date fields do not have to be changed. The **Standards for Year 2000 Conversion / Year 2000 Technical Panel**, **Document E** in the **Reference Library** has more details about compression and encoding techniques.

This is most useful when many, if not most, of the programs using the file do not reference the date field(s). The drawback is that the code to handle compressed fields is more complicated and has to be placed in every program referencing these fields.

2. 4. 2. 3. Field Expansion

With the field expansion choice, new programs, data files and databases must have the four-digit year stored. For legacy systems, field expansion can be used for impacted date elements by expanding 2-digit year date fields defined in working storage, linkage section, and copybooks to 4-digit year variables.

Some programmers use the following technique to allow continued use of the original variable name. It also allows use of the four-digit year when needed.

```
03  ORIGINAL-FIELD-8
      05  ORIGINAL-FIELD-CC          PIC 9(2)
      05  ORIGINAL-FIELD             PIC 9(6)
03  ORIG-FIELD-REDEF REDEFINES ORIGINAL-FIELD-8
      05  ORIGINAL-FIELD-CCYY        PIC 9(4)
      05  FILLER                     PIC 9(4)
```

where **ORIGINAL-FIELD** is the original field name.

In certain situations, such as date comparisons and aging functions, fields must be changed to use the four-digit year. Other techniques and considerations that must be followed include:

- **Remove all existing century derivation logic internal to programs**. Century derivation logic is one type of an ad-hoc solution. Such solutions may be complex and are a maintenance problem. When the year fields have been expanded, the century derivation logic can be removed from existing code.
- **Use Standard Date Handling Subroutines**. Always use the most standard technique available. Date retrieval can be done with COBOL ACCEPT commands and vendor supplied date routines (C$CURDATE and IGZEDT4). Where possible, the TransCentury date package is HUD's standard for converting, comparing and validating dates. Programs written using the older COBOL compiler on the Unisys platform (@ACOB) may not be able to use TransCentury. Contact Team 2000 for information about available routines.
- **Change all hard coded date values**. Examine all hard coded dates used within programs including date literals defined to only have a YY value. Make sure the literals are Year 2000 compliant.
- **Use Standard Date Formats Within an Application**. The application needs to define a standard date format. This same format must be used for every changed field.
- **Screen and Report Layout**. Years displayed on screens and reports may not require expansion. As long as the meaning is clear to the person reading the field, expansion is not needed. This is especially useful when multi-line printouts would have to be modified to align everything.

➤ **Data Entry**. Keystroke volume is an important factor in the cost of data entry. There is no need to expand data entry date fields unless the century is not clear to either the terminal operator or the program.

2. 4. 2. 4. Relative Dates

Another option is to replace the stored date with a number relative to a given base date. After establishing a "day one" all date values are converted to an integer. This simplifies calculations involving the "days between" two dates or the calculation of a date *n* days before or after a specified date. Relative dates also take less space and can be stored in binary integer saving even more room. A disadvantage of relative dates is that individual date elements (month or day) must be computed to be used or displayed, instead of being present by a field definition.

The problem with relative dating is that every computation trusts that the same base day is used. If one base date is used when converting from a date to a number and another is used to reverse the process, the result is chaos. **Table 2-3** shows some of the starting dates in use.

Table 2-3: Relative Dates

Day One	Used By	Comments
January 1, 0001	TransCentury and DB2	
October 15, 1582	IBM Language Environment	This is also known as a Lilian Date (See **Document B**, **Definitions** in the **Reference Library**).
January 1, 1601	COBOL 370 and the ANSI-85 (Extended) Intrinsic Functions	This date has some computational advantages.
January 1, 1900	PC based software such as Lotus 1-2-3	Some of these products incorrectly identify 1900 as a leap year. Make sure you know that year is handled by the software whose date you are reading.

An application could obviously specify any date as the start date. Again, to ensure accurate conversions, "Day One" must be identified and remembered by the entire development and maintenance team.

2. 4. 3. File and Database Expansion

Once the strategy for each field is determined, the strategy for expansion and modification of records, files and databases needs to be examined. The TransCentury package is especially useful for this task. It can handle date validation and date conversion to and from a huge variety of formats. TransCentury can also list the occurrences of invalid data. A purification process is then needed to replace invalid data. This task requires extensive effort from both maintenance and end-user personnel.

2. 4. 3. 1. Data Purification

Over time, values which seem to be invalid may be stored in any field. In some cases these values are just wrong. In others, special values are used to "flag" special cases. For example, zeroes in the Retirement Date would mean the employee has not retired.

Data purification is the process of finding all the suspicious cases, deciding what to do about them, and making needed changes. This process is especially critical in the Year 2000 effort since values may have a different meaning in light of a four-digit year. Time must be allocated to:

- ➤ Read through the files,
- ➤ Determine cases to analyze,
- ➤ Correct invalid data,
- ➤ Determine strategies for handling "flag" fields, and
- ➤ Implement the required coding changes.

2. 4. 3. 2. File Expansion

In some cases, field expansion forces file or record size expansion. This happens when every byte in the record is used. In these cases, the record size must be expanded to allow room for the expanded date.

Many copybooks and record layouts have FILLER fields at the end to allow room for new or expanded field. This expansion factor allows the fields to be expanded without expanding the record size.

Example:

```
01  OLD-REC
    03  SMALL-DATE   PIC X(6).
    03  FILLER       PIC X(20).

01  CHANGED-REC
    03  BIGGER-DATE  PIC X(8).
    03  FILLER       PIC X(18).
```

2. 4. 4. Bridge Programs

Bridge programs simplify the conversion of data files by avoiding the need to convert everything at once. The idea is that a stand-alone program –a bridge–can perform any needed conversions, enabling non-compliant access by reading data in one format and writing it in another. This lets the two systems be converted independently without interrupting data transactions. Bridges should be kept in place until it is mutual agreed that they are no longer needed; that the data being exchanged is both valid and compliant.

Table 2-4 presents sample logic for determining if bridging is necessary and helps select the appropriate kind of bridge.

Table 2-4: Sample Bridging Determination and Selection Matrix

Date Size	Writing Step uses Small Dates	Writing Step uses Large Dates
Reading Step uses Small Dates	No Bridge Needed	Bridge reads large dates and writes small dates
Reading Step uses Large Dates	Bridge reads small dates and writes large dates	No Bridge Needed

2. 4. 5. Database Field Expansion

Database field expansion requires significant coordination and planning. Bridging does not work in this case. The database is in the same format for all users. Every user of a database has to convert at the same time.

Tools exist to aid in the reformatting effort. I-QU 2000 converts Unisys databases and files.

There are different solutions for the other platforms. On the Hitachi, Database 2 (DB2) is already compliant, so it will not have to convert. On the PC/LAN platform we are only concerned with the older COTS (commercial off-the-shelf) databases such as those in FoxPro and older DB2 versions. For these, either upgrade to a compliant, higher version of the same COTS package, or convert the database to a compliant version of another COTS package.

2. 5. Platform Readiness

HUD applications reside on three platforms:

1. Hitachi,
2. Unisys, and
3. Local Area Network (LAN).

Each of these has unique requirements which must be met to enable Year 2000 certification to take place. However, development and testing can take place without a fully compliant platform. For example, development and testing of a Database 2 (DB2) application is not dependent on an Information Management System (IMS) utility.

Software development can continue until it becomes dependent on a platform issue on the critical path or the Year 2000 work plan. The basic issues for each platform are discussed below.

2. 5. 1. Hitachi

There are three Hitachi systems at HUD. The production system **(H)** is used to run HUD's business applications. The test system **(D)** is used to develop and test changes. A new system **(Y)** has been created as a Year 2000 testing platform. All three systems now use OS/390 Release 3, the most current operating system, which is Year 2000 compliant.

Lockheed-Martin provides the facilities for HUD and is responsible for the installation and maintenance of operating system software. A complete list of this software is prepared by Lockheed-Martin and is available on the HUD LANs (use the HUDMSD icon). Lockheed-Martin is working with software vendors—including third-party vendors—to determine the Year 2000 compliance status of these products.

CSG produces Computer Technical Bulletins (CTB) describing upgrade requirements for new releases. These are stored in the SYSA.USER.GUIDE Partitioned Data Set (PDS). The member name is in the form **TECH##**. If the bulletin you are seeking is not there, contact CSG. **Table 2-5** describes a method for retrieving these bulletins:

Table 2-5: Computer Technical Bulletin (CTB) Hitachi System Retrieval Procedures

Retrieval Procedures:
1. Logon the TSO 2. Select Item G on the ISPF Panel 3. Place Asterisk (*) on the topic line (A list of member names will appear) 4. Type an S to the left of the item you wish to see 5. Hit the Enter Key

2. 5. 1. 1. Operating System

The operating systems for HUD's Year 2000 production platform (**Y** system), development platform (**D** system) and production platform (**H** system) were upgraded to IBM's OS/390 Release 3 in 1997. OS/390 is Year 2000 compliant.

2. 5. 1. 2. Transaction and Database Software

HUD must move to recent levels of Customer Information Control System (CICS) and Database products to have compliant software products.

2. 5. 1. 2. 1. CICS Upgrade

CSG has upgraded CICS to the Year 2000 compliant version 4.1 on all HUD systems (Year 2000, Development and Production). In conjunction with this upgrade, DB2 must be upgraded from version 2.3 to version 3 and IMS must be upgraded from version 2.1 to 5.1. The following considerations must be examined:

- ➤ All CICS applications must use CA-Top Secret security features for access control.
- ➤ CICS macro support is removed. Programs using CICS macros may require code changes, some of which are very extensive. This is a major consideration for COBOL programmers and may become a serious problem once the code is identified.
- ➤ Global user exits have been changed.

2. 5. 1. 2. 2. DB2 Upgrade

CSG has upgraded from Database 2 (DB2) version 2.3 to version 3.1, a Year 2000 compliant release.

2. 5. 1. 2. 3. IMS Upgrade

Information Management System (IMS) version 5.1 is a Year 2000 compliant release. The changes needed to application programs and Job Control Language (JCL) to move from IMS 2.1 (the version HUD began with) to IMS 5.1 (the compliant version) are extensive. Major changes occurred in Version 4.1 making it a major bridge version. To allow the applications to cross this bridge, CSG formulated a plan to upgrade from IMS version 2.1 to IMS version 5.1 in two steps:

- ➤ 2.1 to 4.1 (Completed in 1997)
- ➤ 4.1 to 5.1

Extensive changes are required to IMS applications and databases to ensure a Year 2000 compliant structure. The basic issues affecting this upgrade path are as follows:

- ➤ Database control,
- ➤ Security, and
- ➤ Communications.

IMS version 5.1 uses database control (DBCTL) exclusively. All IMS database accesses must be upgraded from DL/I to DBCTL. This can occur in either IMS version 4.1 or 5.1. CSG has installed the DBCTL capability in IMS 4.1 to let applications personnel make the changes in the interim version.

DBCTL processing requires changes in the way IMS databases are handled at HUD. These are major changes. Applications personnel, CICS and system programmers, and IMS database administrators all require additional training. New computer operations procedures must be documented and the operators must be trained.

Changes include:

- ➤ DL/I batch jobs must be changed to BMP (Batch Message Processing),
- ➤ Batch Backout/Recovery procedures must be changed to DBCTL,

- ➤ CICS regions and batch jobs with multiple copies of a database must change,
- ➤ Program changes are needed if I/O PCB does not exit, and
- ➤ All batch update PSBs must change to PROCOPT=E, to avoid locking the DBCTL.

2. 5. 1. 3. COBOL Compilers

Many applications are written in software which is either no longer supported or does not support Year 2000 conventions when processing date features. The safest solution is to migrate all programs to COBOL for OS/390 and execute under the language environment run time libraries. Brief summaries of each version of COBOL follow.

2. 5. 1. 3. 1. OS/VS COBOL

OS/VS COBOL and its libraries are no longer maintained or supported by IBM. There are no plans to add Year 2000 compliant date/time services to this COBOL compiler. Therefore, a request for a date will return a two-digit year and no century.

Some OS/VS COBOL programs might execute properly with Language Environment run-time libraries, but case-by-case testing is required. Migrating to COBOL for OS/390 means moving to a fully supported product. COBOL for OS/390 programs can be mixed with OS/VS COBOL programs.

2. 5. 1. 3. 2. COBOL II

Many of HUD's COBOL programs use the COBOL II compiler. It is possible to execute programs written in COBOL II and make them Year 2000 compliant. Various options are available, such as windowing and calling COBOL for OS/390 subroutines. A brief summary of the main issues is provided below.

- ➤ If any date calculations are needed, use the TransCentury calendar routines. These are tested and are Year 2000 compliant. If the only use of a date is to retrieve the system date for printing or display, the COBOL function IGZEDT4 can be used. IGZEDT4 returns the date as YYYYMMDD.
- ➤ Storing a date as the number of days since a beginning date is known as relative dates. These only work if all date numbers use the same starting date. The Language Environment provides three functions which handle relative dates. They are CEEDATE, CEEDAYS, and CEESECS. These functions use October 15, 1582 as the starting date. Do not mix these with other COBOL relative date functions because they use January 1, 1601 as the starting date.
- ➤ Use of language environment run-time libraries is mandatory.
- ➤ Support for C++ is provided in the language environment OS/390 run time libraries; there is no support for PL/1 callable programs.
- ➤ COBOL subroutines may fail on 4-digit dates when:

- ➤ Sorting by date, or
- ➤ Date calculations are used for duration.
- ➤ COBOL for OS/390 and language environment are not dependent on the operating system for development and testing.
- ➤ If code conversion is required, IBM COBOL CICS Conversion Aid (CCCA) 5785-ABJ can be used to assist in upgrading to American National Standards Institute (ANSI) standard code.

2. 5. 1. 3. 3. COBOL for OS/390

This is the latest version of COBOL and is Year 2000 compliant. While some work-arounds are available for other versions of COBOL, moving to COBOL for OS/390 while making the other Year 2000 changes ensures one upgrade is sufficient to bring a program into compliance.

Reasons to migrate include:

- ➤ Service support for the COBOL compiler,
- ➤ Solve the Year 2000 problem for applications with PL/1 mixed with COBOL under language environment,
- ➤ Use COBOL programs as DB2 Stored Procedures,
- ➤ Use COBOL for CICS version 4 Autoinstall EXIT,
- ➤ To load COBOL programs, COBOL data, or I/O buffer above the 16 megabyte line, and
- ➤ To mix COBOL with C/C++ programs.
- ➤ There are also object-oriented language extensions in IBM COBOL for OS/390. These are based on the emerging International Standards Organization (ISO) and ANSI COBOL standards and are natural syntax extensions to COBOL—not a new language.

OS/390 is the prerequisite run-time environment for applications generated with the following IBM compiler products:

- ➤ COBOL for OS/390
- ➤ SAA AD/Cycle for COBOL/370
- ➤ PL/1 for OS/390
- ➤ AD/Cycle PL/1 OS/390
- ➤ FORTRAN for OS/390
- ➤ OS/390 C/C++
- ➤ IBM C for VM/ESA

Language Environment allows the creation of mixed-language applications. It provides a common set of tools and debugging aids which are accessed by code written in any of the allowable languages. A set of assembler macros is also provided to allow assembler programs to run within Language Environment.

With Language Environment and OpenEdition Distributed Computing Environment (DCE) Application Support, programs can access host data and application logic in Customer Information Control System (CICS) and IMS from any DCE client in the network. Client code is written in C and server code in C, COBOL, or PL/1 to access code you already have. Certain restrictions apply for COBOL and PL/1. Locale callable services enhance the development of internationalized applications.

A complete description of Language Environment's condition handling model and message services is in OS/390 Language Environment Concepts Guide.

2. 5. 1. 4. Security

All security must be handled EXTERNAL (Top Secret) to provide a secure interface to batch BMPs. This is required in CICS version 4.1 as well.

2. 5. 1. 5. Communications

The IMS LU6.1 adapter for LU6.2 applications is removed in version 5.1. Customers are advised to upgrade to APPC/IMS if they are using the LU6.1 adapter.

2. 5. 1. 6. Third Party Software

In some cases, HUD is no longer licensed to use the product. CSG is investigating third party tools, provider upgrade plans and schedules. All of these software products possess dependencies which must be considered.

2. 5. 2. Unisys

The Unisys software on the systems at HUD is Year 2000 compliant. The System Base 6 (SB6) release includes the full range of software from operating systems through database management and compilers. With the exception of LINC applications, this level has been installed with no need for program upgrades. Only a few issues affect this platform. A complete list of software, along with availability and status is available in **the HUD Year 2000 Tools Overview**, **Document F** in the **Reference Library**.

2. 5. 2. 1. LINC Level 16

Some Unisys applications need to be upgraded to LINC Level 16. The upgrade requires changes to the user code. This issue is being handled by applications personnel.

2. 5. 2. 1. 1. Date Storage Format

Does the application store relative or absolute dates? A relative date is a day count from the system's base year (1957) while an absolute date is the month, day, and year in the conventional format.

If the absolute date format was used without the century, the application team may opt to:

- ➤ Convert to relative date format,
- ➤ Expand the existing year field from two digits to four by adding century, or
- ➤ Provide a separate field for the century.

Before converting an absolute date with no century to a relative date format, the system data item **glb.century** must first be initialized to either 0 or 19. Otherwise, **glb.century** remains at 20 if the year of the date converted is less than 57 (base year). The resulting relative date would be incorrect, having the conversion of a date with century 19 coming after the conversion of a date with century 20. Initializing **glb.century** eliminates this problem.

2. 5. 2. 1. 2. Global Logic for Date Conversion

Check whether the system has global logic GP-VALIDATE-DATE and that it is consistently inserted in all absolute date conversions. This is an alternative to initializing **glb.century** to 0 or 19. It validates the date in MMDDYY or MM/DD/YY format and includes setting **glb.century** to proper value.

The validation process translates the date into many other formats, one of which is relative date, which should then be moved to the date field for database storage. The global logic can be copied from S01DB specification. If the date is in YYMMDD format, the application team will have to modify the global logic to suit their requirement.

2. 5. 2. 1. 3. Screen/Report Input and Display

Determine how the end users want the dates to be presented to them on screens or reports. LINC 16 provides 32 different formats for displaying the date. If date with century is preferred, coding for date conversion will be simple, while date without century requires an additional command such as inserting the global logic **GP-VALIDATE-DATE** or setting the **glb.century** to 0 or 19 prior to date conversion. (See Sections **2.5.2.1.1** and **2.5.2.1.2**).

2. 5. 2. 2. Third Party Software

Utilities in this category need to be investigated for compliance. If the software is found to be non-compliant, and is under a maintenance contract, then the vendor will be asked when a Year 2000 compliant version will be available. If the software is not supported, then a suitable replacement must be found, or the software must be dropped from the HUD inventory. **Table 2-6** lists the Year 2000 compliance status of HUD-owned, Unisys OS 2200 SBR6 platform third party software, as of March 31, 1997.

Table 2-6: HUD Unisys OS 2200 Third Party Software

Vendor	Product	Installed Version	Year 2000 Compliant Version	Compliant Version Available	Year 2000 Compliance Announced
Attachmate	DataXpress (file transfer)	3.250			Jan 97
Attachmate / Comp Intel Corp.	FTPLUS (file transfer from PC to system)	2.11.08			Jan 97
Formula Consultants	Star 1100 (magnetic tape resources)	7R1	7R1	Dec 95	Jul 95
KMSystems, Inc.	Infoquest	3R2B	4R1E	Jun 95	Sep 96
	Infoquest	4R1C	4R1E	Jun 95	Sept 96
	IQU PLUS-1	11R2	11R2	Jul 94	Sept 96
	Qlink	6R1	6R2	Jul 94	Sept 96
	Qlink	5R1A	6R2	Jul 94	Sept 96
Marble Computer, Inc.	DCD3	1R1	2.2	Apr 97	Jul 97
P2 Software	Biller 1100 (billing system)	8R1	8R2	Sept 97	June 97
SAS Institute	System 2000 (S2K)	4.0			TBD
SMA	Scheduler	6R1S	7R1	Jan 98	Feb 97
SPSS, Inc.	SPSS-X	2R0A			TBD
Systar, Inc.	Systar Aladdin (SystemVision)	7R2A	8R2	Jun 97	Jan 97
	Systar Boris (SystemVision)	7R2A	8R2	Jun 97	Jan 97
	Systar Focus (SystemVision)	7R2A	8R2	Jun 97	Jan 97
	Systar I/O (SystemVision)	7R2A	8R2	Jun 97	Jan 97
	Systar Olga (SystemVision)	7R2A	8R2	Jun 97	Jan 97
Team Quest	MS Manager	5R3A	5R3A	Aug 97	Oct 97

(As of January 5, 1998)

2. 5. 2. 3. Unsupported Products

There are several categories of products which are not supported but are being used on the Unisys 2200 platform. These include:

- **Shareware** - Unisys (and, previously, UNIVAC) users exchanged utility programs. There was no charge for these tools and no promise to maintain them.
- **Category 3** - Several products were originally developed by Unisys but have been unsupported for years. These are in maintenance category 3. Sites can use them, but do so at their own risk.
- **Miscellaneous authors** - Other products are no longer supported. Some were written at HUD. Others were obtained for other sources. Even if they were purchased, the vendors have either gone out of business or stopped supporting the products.

The HUD Year 2000 Tools Overview, **Document F** in **the Reference Library**, contains a table showing the Year 2000 compliance status of these products.

2. 5. 3. Local Area Network

The Year 2000 environment is a segment of the HUD Local Area Network (LAN). The Year 2000 Test LAN is composed of Year 2000 compliant servers, workstations and network operating system.

2. 5. 3. 1. Operating Systems

The Year 2000 LAN platform's base system is the Novell 4.11 network operating system, which is compliant. HUD's other LANs currently use Novell 3.x base systems, which are not. These systems are expected to either be upgraded or replaced. HUD is currently evaluating the Novell 4.11 and Windows NT 4.0 operating systems to determine which will be the base system for the Year 2000. Novell 4.11 and Windows NT have been certified by IT as compliant with the date changes in the Year 2000. The Windows NT operating system is utilized solely as the platform upon which the database systems operate and communicate on the network. Windows NT 5.0 is not expected to be available until early 1998 and therefore would not meet the Y2K schedule requirement. HUD has some IBM OS/2 and Sun Solaris platforms which must be reviewed for compliance.

2. 5. 3. 2. HUDware II

The Year 2000 test environment has been set up on the new HUDware II platform, the upgrade of HUDware 2.0. HUDware II provides a 32-bit operating system with preemptive multitasking, improved error recovery, and more power for complex math computations. All HUD Windows-based and DOS applications have been configured and tested to run on this platform.

HUDware II software, as currently defined, is not totally compliant with Year 2000, however. Some of the software uses windowing, while others are compliant only if the application was written correctly.

HUDware support files for HUD applications reside in a dedicated volume on the LAN servers.

2. 5. 3. 3. Client/Server Environment

HUD's Client/Server applications are those which divide their processing power between a workstation and a dedicated database server. The database server is where the database structures of the application reside. From the client workstation, information on the database server can be retrieved, modified, and stored.

Three database servers have been established by Team 2000 to provide a test-bed environment for the compliance testing of HUD's database applications. This test-bed mirrors the established HUD environment. These database servers –Microsoft SQL, Sybase, and Lotus Notes—use Windows NT version 4.0 as the operating environment upon which they run and communicate on the network.

The specifications are as follows:

➤ **Microsoft SQL Server specification**
- Microsoft Windows NT 4.0, Service Pack 3
- MS SQL version 6.5, Service Pack 3
- 1GB Memory
- 85GB Hard Disk Space
- Compaq Proliant 5000R

➤ **Sybase Server specification**
- Microsoft Windows NT 4.0
- Sybase System 11
- 512 MB Memory
- 16GB Hard disk space
- Compaq 4500

➤ **Lotus Notes Server specification**
- Microsoft Windows NT 4.0
- Lotus Domino 4.5, Release 4.5.2
- 32 MB Memory
- 2GB Hard disk space
- Dell OptiPlex GXMT 5133

2. 5. 3. 4. Middleware Compliance Report

Middleware mediates the Client/Server link and smoothes out the potential incompatibilities between communication protocols, database query languages, application logic, and operating systems. It interconnects legacy computer systems and opens them up to today's modem technologies of workstations, LANs, and SQL servers. Database applications use some form of middleware, along with such *de facto* standards as Open Database Connectivity (ODBC). HUD has tested and implemented a number of standard Client/Server infrastructure components, which are compliant with the impending date changes at the approach of Year 2000. Non-compliant components will be upgraded to the compliant version before Year 2000. For a list of Middleware currently being utilized, see **Table 2-7** below.

Table 2-7: Year 2000 Middleware Product Compliance List

Product	Version	Manufacturer	Compliant?	Upgrade	Function
MDI Gateway	2.05.02	Sybase, Inc.	Yes	Direct Connection 10.5	Allows developers to create C/S applications that directly access DB2 and other data sources residing on the Hitachi mainframe.
InfoPump Idb Server	1.02.00	Platinum Technology, Inc.	TBD	Version 3.0	Transforms data type from one vendor's database into different data type for another vendor's database .
InfoPump Manager	2.20.00	Platinum Technology, Inc.	TBD	TBD	Moves data between disparate systems.
InfoHUD	1.3.0	Platinum Technology, Inc.	TBD	Version 2.0	Presents a relational database interface to non-relational legacy data on HDS mainframes.
OmniCONNECT	10.5	Sybase, Inc.	Yes	Version 11.5	Provides a unified view to heterogeneous databases.
Open Client	10.0.3	Sybase, Inc.	Yes	Version 11.1.1	Connectivity software for Sybase servers.
Impromptu	3.0/3.5	Cognos, Inc.	Yes	Version 4.0	Database query tool.
Data Shopper/MVS	1.3.11	Platinum Technology, Inc.	TBD	TBD	Windows-based GUI tool to access metadata in repository.

2. 5. 3. 5. LAN Certification Test Guidelines

To insure prompt and proper certification testing on the Y2K LAN the following guidelines are provided:

- ► Dates for testing are provided on the Year 2000 Integrated Implementation Plan.
- ► A pre-certification meeting is held by TEAM 2000 Certification Team to review SEG Test Workplan and discuss issues regarding testing on the Year 2000 LAN Platform.

- ➤ A Year 2000 LAN Test Migration Request form would be filled out and submitted electronically, identifying Platform testing resources required.
- ➤ LAN Branch and Team 2000 LAN Coordinator will make resources available to perform certification testing.
- ➤ Due to the sensitivity of data, and pre-existing restriction on HUD network, access to the Year 2000 Test Server will be made available to testers only for the movement of all necessary system files from current location to the Y2K Test Server.
- ➤ Date changes are made on servers as needed during testing. For Client/Server applications, the date/time manipulation procedures are crucial. Developers should state what type of date/time manipulation procedures are to be performed.

The Year 2000 LAN Test Migration Request form should be forwarded, electronically, to the Representatives for each division. Forward a carbon copy (cc) to the points of contacts associated with the platform. Submit Test Migration Request form, at least, 10 business days prior to desired test date.
A sample Test Migration Request form is located in **Chapter 5 (Testing)**, **Figure 5-2**.

2. 6. Develop Workplan

The information gathered in this stage allows for the development of a detailed workplan; including information about the changes to be made. There is also scheduling and coordination information to consider.

Update the plan when changes occur. An updated plan provides a method of communication among the team members and becomes a tool to reduce the negative impact of turnover. Renovators and management will also have a better sense of the project's progress. **Table 2-8** presents a sample Year 2000 project workplan.

Once the workplan is developed, enter the information into the STATUS 2000 database. This will allow for easy updating and tracking of the plan by all involved in the project. See **STATUS 2000 Database Information**, **Document I** in the **Reference Library**, for user information.

Table 2-8: Sample Year 2000 Project Workplan

Task	Work Days	Duration	Start Date	End Date	Resource
Application Expert Guidance	30	50	1/6/97	3/14/97	E1
Team Leader Oversight	48	50	1/6/97	3/14/97	T1
Establish Baseline	5	5	1/6/97	1/10/97	T1
Application Review	20	10	1/6/97	1/17/97	S1, S2
Determine High-Level Approach	5	5	1/13/97	1/17/97	T1
Develop Work Plan	5	5	1/13/97	1/17/97	T1
Develop Design Code Changes	8	4	1/21/97	1/24/97	S1, R1
Environment Update Decisions	4	4	1/21/97	1/24/97	R2
Develop Test Plan ➤ Develop Test Data	32	9	1/21/97	1/31/97	S2, R3, R4, R5
Make Code Changes / Unit Test ➤ Develop Bridges / Unit Test ➤ Develop Conversion Programs / Unit Test ➤ Database Redesign / Unit Test	70	15	1/27/97	2/14/97	S1, R1, R2
System Integration Test ➤ Regression Test ➤ User Acceptance Test	59	15	2/10/97	2/28/97	
Perform Conversion / Implementation	5	5	3/3/97	3/7/97	
Update Documentation	5	5	3/3/97	3/7/97	
Revisit Year 2000 Process	10	10	3/3/97	3/14/97	
Total: 306 workdays spanning 50 days					

Person Code	Role	Assumptions
E1	Application Expert (Task Leader)	➤ Holidays and weekends are not workdays. ➤ No vacation time is used. ➤ Application Expert spends 2/3 of their time on Year 2000. ➤ Technical environment is completely established by January 6, 1997.
T1	Team Leader	
S1	System Analyst 1	
S2	System Analyst 2	
R1	Renovator Tester 1	
R2	Renovator Tester 1	
R3	Renovator Tester 1	
R4	Renovator Tester 1	
R5	Renovator Tester 1	

(This page has been left blank intentionally.)